



APRESENTAÇÃO

Em um sistema computacional, a memória é representada pelos dispositivos que permitem a um computador armazenar dados, de forma temporária ou permanente. Dessa forma, pode-se dizer que "memória" é um termo genérico para nomear componentes de um sistema, capazes de armazenarem dados e programas. Há vários tipos de memórias em um mesmo sistema computacional, uma vez que cada um desses tipos de memória possui características e especificidades que normalmente determinam sua utilização específica no sistema.

Nesta Unidade de Aprendizagem, você verá o conceito de memória, a alocação das memórias estática e dinâmica e o modo como essa alocação é influenciada pelas suas características. Verá também como identificar os diversos tipos de frequências em que as memórias trabalham, e como esta frequência influencia no desempenho.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Descrever as características de desempenho das memórias.
- Diferenciar as características de alocação das memórias estática e dinâmica.
- Identificar os tipos de frequências de memórias e suas especificidades.



INFOGRÁFICO

As memórias não voláteis são de suma importância para a computação, uma vez que são responsáveis por armazenar os sistemas e os seus dados, bem como os dados dos usuários.

Na história da computação, diversos modelos de memórias não voláteis surgiram e desapareceram do mercado, em uma linha evolutiva dos parâmetros de análise das memórias, proporcionando um ganho de performance em todas as áreas desses desempenhos.

Veja, no Infográfico, a evolução das memórias não voláteis, algumas das tecnologias já utilizadas e ainda em uso na atualidade.



CONTEÚDO DO LIVRO

Desde seu surgimento, os sistemas computacionais, em geral, fazem uso de uma variedade de tipos de memórias, que, conforme suas especificidades, contribuem para o sistema de diferentes formas.

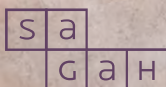
A forma com que o programa ou sistema utiliza as memórias é influenciada pelo tipo. Por exemplo, as memórias com fita magnética, apesar de serem muito lentas, ainda são muito utilizadas porque são baratas. Dessa forma, tendem a ser utilizadas em sistemas de *backup*, onde a performance não é um fator limitador, mesmo que os *backups* ocupem muito espaço de armazenamento. Logo, uma memória que disponibiliza tal espaço com custo mais acessível se torna ideal para esse tipo de operação.

No capítulo, Parâmetros de Análise das Memórias, do livro, *Arquitetura e Organização de Computadores*, você verá as características de cada tipo de memória e suas formas de alocação, bem como a alocação influenciada pelas diferentes memórias que um determinado sistema computacional pode disponibilizar. Verá, ainda, os tipos de frequências em que as memórias podem trabalhar, e como estas influenciam na performance.

Boa leitura!

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Diego Bittencourt de Oliveira



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Parâmetros de análise das memórias

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever as características de desempenho das memórias.
- Diferenciar as características de alocação das memórias estática e dinâmica.
- Identificar os tipos de frequências de memórias e suas especificidades.

Introdução

Existem diferentes tipos de memórias em um mesmo sistema computacional e suas diversas características influenciam na sua aplicação. Por exemplo, temos memórias muito rápidas desde os primórdios da computação, que são as memórias nas quais são criados os registradores. Porém, sua fabricação é extremamente cara, de modo que apenas pequenas quantidades das mesmas eram disponibilizadas praticamente junto do núcleo de processamento dos microprocessadores.

Neste capítulo, portanto, você vai conhecer os parâmetros de análise das memórias e a forma de utilização dessas memórias pelo programa ou sistema que as estará utilizando. Além disso, você vai ver, também, os conceitos relacionados às frequências que as memórias podem possuir e em que situações essas frequências podem influenciar o seu desempenho.

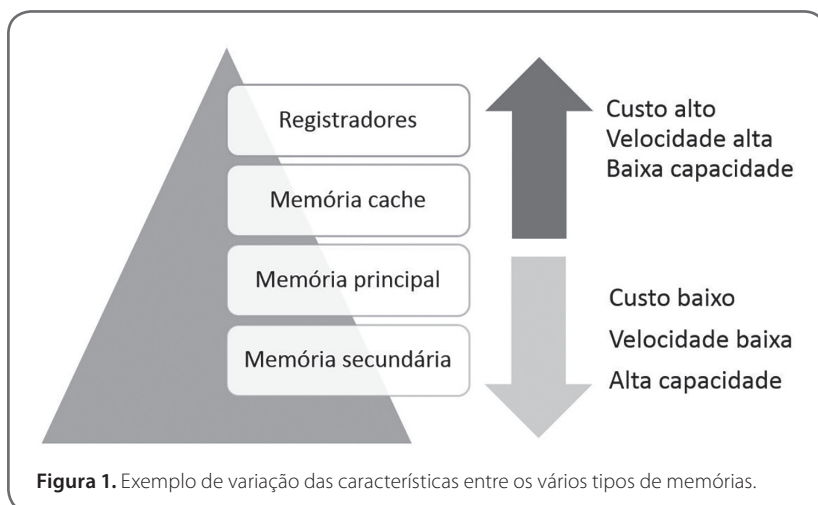
Características de desempenho das memórias

Segundo Monteiro (2007), objetivando uma correta manipulação das informações, sejam instruções ou dados na memória de um computador qualquer, observa-se a necessidade de que, em um mesmo computador, haja diferentes tipos de memória. Para determinadas funcionalidades, é fundamental que a

transferência de informações seja realizada da forma mais rápida possível. As atividades realizadas no núcleo de um microprocessador são um exemplo no qual a velocidade é fundamental, mas a quantidade de bits a ser manipulada é muito pequena.

Assim, temos a caracterização de tipos de memória diferentes, em que, em determinados casos, a capacidade de armazenamento é mais importante que a velocidade pela qual a informação vai trafegar entre os componentes do sistema. Em virtude da grande variedade de tipos de memória, não é possível desenvolver um sistema de computação com uma única memória. Existem muitas memórias no computador que se interligam de forma estruturada, constituindo um sistema em si, parte do sistema global de computação, que pode ser denominado subsistema de memória.

O subsistema citado é planejado de modo que seus componentes sejam organizados de forma hierárquica, conforme mostrado na estrutura em forma de pirâmide que pode ser observada na Figura 1. Vale observar que o formato de pirâmide desta estrutura de memórias não é uma escolha qualquer, uma vez que a pirâmide permite que observemos a variação de crescente de características que cada tipo de memória possui.



A variação de características citadas está diretamente ligada às características de desempenho de cada tipo de componente. Essa variação de características se dá com base nas características de alguns parâmetros. Os principais parâmetros para análise dessas características são:

- tempo de acesso;
- capacidade;
- volatilidade;
- tecnologia de fabricação;
- temporariedade;
- custo.

Segundo Null e Lobur (2011), o tempo de acesso é uma característica de desempenho que indica quanto tempo a memória gasta para transmitir uma informação no barramento de dados após uma determinada posição de memória ter sido endereçada — ou seja, o período de tempo desde o momento em que foi iniciada a operação de acesso até que a informação solicitada (instrução ou dado) tenha sido efetivamente transferida. Esse é um dos parâmetros que pode medir o desempenho da memória e que pode ser nomeado de tempo de acesso para leitura ou apenas tempo de leitura.

O tempo de acesso de uma memória depende do modo como o sistema de memória é fabricado e da velocidade dos seus circuitos. Essa característica varia muito de acordo com o tipo de memória, sendo valores típicos atuais aqueles numa faixa entre 50 e 150 nanossegundos (ns), no que se refere à memória principal, e de 12 a 60 milissegundos (ms) para memória secundária, enquanto fitas magnéticas possuem um tempo de acesso na ordem de poucos segundos.



Fique atento

Um detalhe deve ser observado no que diz respeito ao tempo de acesso de diferentes tipos de memórias: as ditas memórias eletrônicas, do tipo RAM, ROM ou *flash*, possuem um tempo de acesso que não varia independentemente da distância física, entre o local de acesso e o local do próximo acesso a ser realizado. Porém, em dispositivos eletromecânicos, discos, fitas, dentre outros, o tempo de acesso pode variar de acordo com a distância física dentre os dados.

A capacidade é talvez a característica mais simples das memórias e se refere à quantidade de informação que pode ser armazenada em uma memória. A unidade de medida mais comum é o byte, mas também podem ser encontradas outras unidades, como células (no caso de memória principal ou cache), setores (no caso de discos) e bits (no caso de registradores). Na Figura 2, você pode ver um dispositivo de memória *flash* com interface USB (conhecida interface que permite a conexão de diversos dispositivos) com capacidade de armazenamento de 4 G bytes, ou seja, 4 bilhões de bytes.



Figura 2. Exemplo memória com a indicação de sua capacidade.

Fonte: Maxx-Studio/Shutterstock.com.

Segundo Hennessy e Patterson (2014), uma memória pode ser do tipo volátil ou não volátil, característica chamada de volatilidade. Uma memória não volátil é a memória que retém a informação armazenada quando a energia elétrica é desligada. Já a memória volátil é aquela que perde a informação armazenada quando a energia elétrica desaparece, ou seja, quando o equipamento é desligado ou até mesmo na interrupção do fornecimento de energia.

Em alguns casos, é possível manter a energia em uma memória originalmente não volátil por meio do emprego de baterias. Esse recurso é muito utilizado em computadores pessoais para armazenar configurações de um computador pessoal, além de manter alguns recursos ativos da placa mãe, como o seu relógio. Na Figura 3, você pode observar uma placa mãe de um

computador qualquer, em que é possível observar no detalhe uma bateria sendo instalada. Ao remover ou quando a carga da bateria for extinguida, as informações retidas serão perdidas.

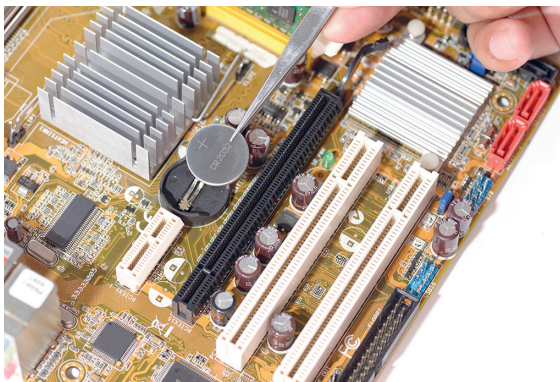


Figura 3. Exemplo de uso da bateria para retenção de informações em memória.

Fonte: Anake Seenadee/Shutterstock.com.

Ao longo do tempo, diversas tecnologias de fabricação vêm sendo desenvolvidas para a fabricação de memórias. Algumas das tecnologias mais conhecidas e utilizadas são as descritas a seguir.

- **Memórias de semicondutores:** são fabricadas com circuitos eletrônicos baseados em semicondutores. Esse tipo de memória é rápido e possui um preço relativamente elevado se comparado a outros tipos. Nessa categoria, existem várias tecnologias específicas, cada uma com suas vantagens, desvantagens, velocidade, custo, dentre outras. Registradores e memória principal são exemplos de memórias de semicondutores.
- **Memórias de meio magnético:** são dispositivos, como os disquetes, discos rígidos (*hard disks*, mais conhecidos como “HDS”) e fitas magnéticas, fabricados de modo a armazenar informações sob a forma de campos magnéticos. Em virtude da natureza eletromecânica de seus componentes e da tecnologia de construção em comparação com memórias de semicondutores, esse tipo é mais barato e permite

o armazenamento de grande quantidade de informação. O método de acesso às informações armazenadas em discos e fitas é diferente; logo, tem tempos de acesso diversos. As fitas magnéticas, normalmente, são memórias com um custo benéfico muito interessante apesar do seu baixo desempenho. Na Figura 4, podemos observar uma fita magnética sendo utilizada em um servidor para armazenar um *backup*, sendo esse o uso mais comum para este tipo de memória.



Figura 4. Exemplo de uso das fitas magnéticas.

Fonte: Kjetil Kolbjornsrud/Shutterstock.com.

A temporariedade é a característica que indica o conceito de tempo de permanência da informação em um dado tipo de memória. Normalmente, as memórias não voláteis possuem um tempo de retenção da informação, ou seja, uma vez que não são utilizadas, a informação armazenada ainda vai estar íntegra até uma quantidade indicada de tempo. Por exemplo, um disco rígido pode vir com a especificação de que possui um tempo de retenção de 15 anos, ou seja, passados 15 anos, as informações armazenadas no mesmo podem já não estar 100% íntegras.

Os registradores, por exemplo, armazenam um dado por um tempo extremamente curto (nanossegundos), o suficiente para o dado ser, em seguida, transferido para a núcleo do processador. Os registradores podem também reter o dado armazenado para um processamento posterior; ainda assim, essa retenção não dura mais do que o tempo de execução do programa ou de parte dele, caracterizando-se como uma memória transitória. Ainda há outros

exemplos de memórias de permanência transitória de dados, como a memória cache e a memória principal, apesar de os dados permanecerem armazenados por mais tempo do que nos registradores.

O custo de fabricação de uma memória, apesar de não se tratar de uma característica de desempenho, influencia os tipos de memórias utilizadas na confecção de um sistema computacional qualquer. Esse custo é bastante variado em virtude de diversos fatores, nos quais podemos elencar como principal a tecnologia de fabricação, que resulta em um maior ou menor tempo de acesso, ciclo de memória, quantidade de bits em certo espaço físico, entre outros.

Uma boa unidade de medida de custo é o preço por byte armazenado em vez do custo total da memória em si. Isso se deve às diferentes capacidades que as memórias possuem; logo, ao calcular o custo pelo preço da memória em si, elas serão naturalmente diferentes, mas dividir o valor pago pela quantidade pela capacidade de armazenamento (o byte) torna esse parâmetro mais real.

Características de alocação das memórias estática e dinâmica

Weber (2012) evidencia que a alocação estática de memória é a mais simplória das alocações, uma vez que ela ocorre durante a compilação de um determinado programa, visto que é esse o momento em que se define uma variável ou estrutura, bem como o tipo e o tamanho da mesma. Na alocação estática, ao executar um determinado programa, a memória necessária para atender às variáveis e estruturas estáticas é reservada e somente fica disponível para utilização quando o programa em questão for fechado ou sua execução concluída.

Na Figura 5, você pode observar cinco exemplos de declaração de variáveis na linguagem de programação C, sendo que essas variáveis serão alocadas de forma estática na ocorrência da execução do programa. Para compreender melhor o espaço em memória que cada variável vai ocupar, observe a descrição de cada exemplo a seguir.

- **Exemplo 1:** o espaço de memória alocado de forma estática para um valor do tipo `char` será de um byte.
- **Exemplo 2:** neste caso, a variável do tipo `int` ocupa um espaço de 4 bytes em memória.
- **Exemplo 3:** no exemplo, temos um vetor com 10 posições de variáveis do tipo `int`, que, conforme o exemplo 2, ocupa 4 bytes de memória e, com 10 posições, ocupa, no total, 40 bytes alocados da memória.

- **Exemplo 4:** neste exemplo, temos uma matriz que, ao total, aloca em memória 9 variáveis (3 x 3) do tipo `double`, que ocupa 8 bytes em memória e, dessa forma, o total de memória alocada é de 72 bytes.

```
// Exemplo 1
char Caractere = 'C';

// Exemplo 2
int Inteiro = 10;

// Exemplo 3
int Vetor[10];

// Exemplo 4
double Matriz[3][3];
```

Figura 5. Formas de alocação estática de memória na linguagem C.



Fique atento

Os tamanhos ocupados em memória por cada tipo de variável, citados no exemplo da Figura 5, estão relacionados à linguagem de programação C. Em outras linguagens de programação, como Java, C#, PHP, dentre inúmeras outras, tais tamanhos ocupados em memória pelos diversos tipos de variáveis que as linguagens citadas possuem podem ser significativamente diferentes.

Dessa forma, em sistemas computacionais nos quais a disponibilidade de memória pode ser um elemento crítico do sistema, é interessante que o desenvolvedor observe tais características da linguagem a ser utilizada no projeto em questão. Assim, é possível a utilização de memória mais eficaz possível conforme a necessidade do projeto.

Assim, pode-se afirmar que a alocação estática de memória não nos proporciona a flexibilidade necessária para solucionar os diversos tipos de problemas que podem vir a surgir — principalmente os problemas nos quais não está claro quantos e quais recursos e variáveis de memória devem ser alocados pelo programa em questão.

Na alocação dinâmica, o espaço de memória do qual as variáveis de um determinado programa vão fazer uso durante a execução do programa é definido enquanto o programa já se encontra em execução — ou seja, quando não se tem ideia ao certo de quanto de memória será necessário para o armazenamento das informações, uma vez que potencialmente as mesmas podem ser determinadas de acordo com a demanda, em tempo de execução conforme a demanda do programa.

Desse modo, a alocação dinâmica ocorre em tempo de execução; sendo assim, as variáveis e estruturas são declaradas sem a necessidade de se definir seu tamanho, pois memória alguma será reservada ao colocar o programa em execução. Assim, durante a execução do programa, no momento em que uma variável ou parte de uma estrutura precise ser utilizada, sua memória será reservada e, no momento em que não for mais necessária, deve ser liberada. Esse processo é realizado utilizando comandos ou funções que permitam, por meio do programa, reservar e/ou liberar memória.

Na biblioteca `stdlib.h` da linguagem de programação C, temos algumas funções utilizadas na alocação de memória de forma dinâmica, sendo as duas funções a seguir as mais utilizadas:

- `malloc()` — função de aloca uma determinada quantidade de memória de forma dinâmica.
- `free()` — função que libera para o sistema uma determinada região de memória alocada pelas funções de alocação de memória da linguagem.

Na Figura 6, podemos observar alguns exemplos de uso das funções e alocação de memória da linguagem de programação C citadas. Veja, a seguir, a função que cada exemplo realiza.

- **Exemplo 1:** temos um ponteiro chamado `vetor` que vai receber o endereço de memória de uma região de memória contendo 100 bytes que foram alocados de forma dinâmica por meio da função `malloc(100)`.
- **Exemplo 2:** neste caso, temos um exemplo um pouco diferente, no qual obtemos o tamanho em bytes da variável do tipo `int` a partir da função `sizeof`, que retorna o número 4 (uma vez que uma variável do tipo `int` ocupa bytes em memória), e multiplicamos esse valor por 100, ou seja, a função `malloc` vai alocar 400 bytes de forma dinâmica da memória. Assim, o ponteiro `vetor` será um vetor de 100 posições de variáveis do tipo `int`.

- **Exemplo 3:** similar ao Exemplo 2, com a diferença de que, após a alocação da memória, é realizado um teste no ponteiro `vetor`, com o intuito de identificar se a alocação foi realizada com sucesso. Caso o sistema não possua a quantidade de memória requerida disponível, o programa não poderá continuar em execução (sendo finalizado pelo comando `exit(1)` após enviar uma mensagem para o usuário indicando a falta de recursos disponíveis com a função `printf`).
- **Exemplo 4:** neste exemplo, a memória alocada no ponteiro `vetor` é liberada para o sistema por meio da função `free`.

```
// Exemplo 1
int *vetor;
vetor = malloc(100);

// Exemplo 2
vetor = (int *) malloc (100*sizeof(int));

// Exemplo 3
vetor = (int *) malloc (100*sizeof(int));
if (vetor == NULL) {
    printf ("Não há memória suficiente para alocação");
    exit(1);
}

// Exemplo 4
free(*vetor);
```

Figura 6. Formas de alocação dinâmica de memória na linguagem C.

De acordo com Stallings (2017), uma das principais diferenças é que na alocação estática não é possível alterar o tamanho do espaço de memória que foi definido durante a compilação, enquanto na alocação dinâmica esse espaço pode ser alterado dinamicamente durante a execução. Na alocação estática, temos a vantagem de manter os dados organizados na memória, dispostos lado a lado de forma linear e sequencial, o que facilita, assim, a sua localização e manipulação. Em contrapartida, precisamos estabelecer previamente a quantidade máxima necessária de memória para armazenar uma determinada estrutura de dados.

Na Figura 7, você pode ver um exemplo de como os dados ficam dispostos na memória do computador, em que temos um vetor de 5 posições de variáveis do tipo `int` posicionadas de forma sequencial. Observe o endereçamento da variável que armazena o valor “10”, que fica na posição “22FF20” (o número do endereço está na base hexadecimal), sendo que a variável que armazena o valor 50 está na posição “22FF24”, que fica a 4 bytes de distância da posição da primeira variável do vetor (ou seja, ela ocupa as posições “22FF20”, “22FF21”, “22FF22” e “22FF23”, visto que a variável do tipo `int` ocupa 4 bytes de espaço em memória).

```
int vetor[5] = {10, 50, 35, 45, 60};
```

22FF20	22FF24	22FF28	22FF2C	22FF30
10	50	35	45	60

Figura 7. Exemplo de dados posicionados em memória alocada de forma estática.

Conforme já mencionado, a quantidade de memória alocada de forma estática é definida durante a compilação, portanto, temos possibilidade de ela sub ou superestimar a quantidade de memória alocada. Nesse caso, esse problema não acontece na alocação dinâmica, visto que a alocação é realizada durante a execução e, assim, sabemos exatamente a quantidade necessária, o que permite otimizar o uso da memória. Além disso, proporciona flexibilidade, uma vez que podemos carregar em memória uma determinada informação que pode ser variável, como um arquivo texto, que pode possuir poucos bytes ou até mesmo milhões de palavras ocupando uma quantia grande de memória.

Tipos de frequências de memórias

De acordo com Delgado e |Ribeiro (2017), a *performance* dos processadores ou dos barramentos de dados é refletida em sua frequência de operação. Normalmente, as memórias possuem a sua velocidade mensurada em nanossegundos (que podemos abreviar com “ns”). Em uma comparação com o intuito de compreender a magnitude dessa velocidade indicada, observa-se

que a velocidade da luz no vácuo é de aproximadamente 300 mil quilômetros por segundo. Dessa forma, é possível observar que, em um nanossegundo, um raio de luz é capaz de percorrer no período indicado aproximadamente apenas 29,98 centímetros.



Fique atento

A diferença entre bit e byte é um conceito importante de ser lembrado, uma vez que alguns periféricos possuem a capacidade de armazenamento indicada em Kb (kilobits) ou em Mb (megabits); já em outros, ela é dada em KB (kilobytes) ou em MB (megabytes). Obviamente, a capacidade do dispositivo varia, de modo muito significativo, quando se usa uma ou outra unidade de medida (1 MB, ou 1.048.576 bytes, equivale a 8 Mb ou 8.000.000 bits). Os prefixos kilo, mega e giga indicam fatores de 1.000, 1.000.000 e 1.000.000.000 quando utilizados para bits. Para bytes, seu significado é modificado para fatores equivalentes a 1.024, 1.048.576 e 1.073.741.824.

No intuito de comparar a velocidade da memória (tempo gasto em cada ciclo) com a frequência de `clock` (número de ciclos que podem ser executados por segundo), é necessário realizar um cálculo, no qual se faz a divisão de 1 segundo pela frequência. Ao realizar tal cálculo, é possível verificar que os seus resultados denotam a relação existente entre MHz e ns.

Na Figura 8, você pode observar alguns exemplos de demonstram a relação citada, em que, na ocorrência do aumento da frequência de relógio, o tempo gasto por ciclo diminui. Logo, a frequência do microprocessador não determina a velocidade que a memória deve ser capaz de suportar. Por exemplo, um microprocessador Pentium de 200 MHz não necessita utilizar um modelo de memória com a velocidade de 5 ns indicados na Figura 8.

No caso do processador exemplificado, a memória utilizada deve conectar-se com o barramento do sistema a 66 MHz, estabelecendo, assim, uma velocidade mínima da memória na casa dos 15 ns para, dessa forma, evitar tempos de espera. Porém, calcular a velocidade ótima da memória adequada para uma determinada frequência de comunicação não é tão fácil como pode fazer supor o exemplo.

Frequência de relógio	Tempo por ciclo	Frequência de relógio	Tempo por ciclo
33 MHz	30,0 ns	166 MHz	6,0 ns
40 MHz	25,0 ns	180 MHz	5,5 ns
50 MHz	20,0 ns	200 MHz	5,0 ns
60 MHz	16,0 ns	233 MHz	4,2 ns
66 MHz	15,0 ns	250 MHz	4,0 ns
75 MHz	13,0 ns	300 MHz	3,3 ns
80 MHz	12,0 ns	333 MHz	3,0 ns
100 MHz	10,0 ns	350 MHz	2,8 ns
120 MHz	8,3 ns	400 MHz	2,5 ns
133 MHz	7,5 ns	450 MHz	2,2 ns
150 MHz	6,6 ns	500 MHz	2,0 ns

Figura 8. Relação entre a frequência de relógio (MHz) e tempo gasto por ciclo (ns).

Fonte: Adaptada de Nóbrega ([201-?]).

Na Figura 9, você pode observar um exemplo real de uma memória comumente encontrada em computadores da sua geração chamada de memória “DIMM”. A indicação de sua frequência é obtida pela inscrição “PC100”; dessa forma, é possível identificar que tal memória é capaz de trabalhar com uma frequência de 100 MHz, ou seja, se a inscrição fosse “PC133” ou “PC166”, teríamos memórias que trabalhariam com frequências de 133 MHz e 166 MHz, respectivamente.

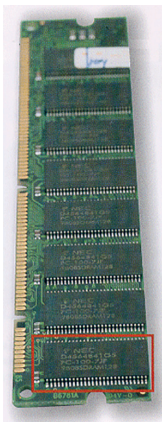


Figura 9. Exemplo de memória real com velocidade de 100 MHz.

Fonte: Nóbrega ([201-?]).

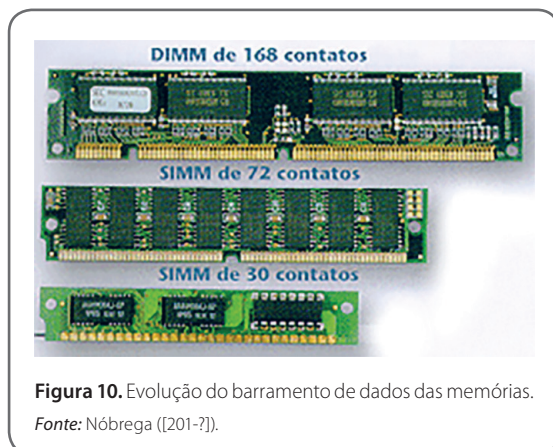
O processo no qual a memória vai realizar a transferência de um dado é dividido basicamente em duas fases:

- **primeira fase:** a posição da memória é localizada, fornecendo, assim, as coordenadas dentro da grade em que se dispõem as células de informação;
- **segunda fase:** a informação é verdadeiramente transferida.

O tempo consumido durante a primeira fase (relacionada à localização do endereço de memória) é conhecido como latência. O tempo real de acesso à memória é o resultado da soma da latência com o tempo por ciclo. Um exemplo seria um módulo de memória, com a indicação de um tempo de acesso de 60 nanossegundos; logo, neste exemplo, há uma latência de aproximadamente 25 ns e de um tempo por ciclo de cerca de 35 ns.

O aumento de frequência dos barramentos de dados e dos microprocessadores contribuiu com o contínuo surgimento de memórias que utilizam técnicas diferentes para atingir acessos de memória muito mais rápidos (NÓBREGA, [201-?]).

Porém, não foi somente o parâmetro relacionado à velocidade das memórias que aumentou, o barramento de dados dessas memórias aumentou exponencialmente na medida em que se necessitou de uma maior velocidade de acesso à memória. Na Figura 10, podemos observar essa evolução, em que os contatos de conexão das memórias aumentaram exponencialmente de geração para geração. A memória “DIMM” possui mais que o dobro de contatos que a sua antecessora a memória “SIMM”.



Este maior barramento possibilita a entrega de uma maior quantidade de dados de uma só vez. A seguir, podemos observar a capacidade de transferência de dados das memórias ilustradas pela Figura 10.

- **SIMM de 30 contatos:** barramento de 8 bits de dados, ou seja, era transmitido de uma só vez 1 byte inteiro pelo barramento indicado.
- **SIMM de 72 contatos:** barramento de 32 bits de dados, ou seja, eram transmitidos de uma só vez 4 bytes inteiros pelo barramento indicado.
- **DIMM de 168 contatos:** barramento de 64 bits de dados, ou seja, eram transmitidos de uma só vez 8 bytes inteiros pelo barramento indicado.

Um maior barramento aliado a maiores frequências das memórias possibilitou que as memórias aumentassem sua capacidade de transferência de dados de forma muito significativa (apesar de elas ainda na atualidade serem um gargalo nos sistemas computacionais). Apesar de esses barramentos mais longos inserirem complexidade nos circuitos a serem desenvolvidos, fornecem uma *performance* em que tal complexidade se torna compreensível.



Referências

- DELGADO, J.; RIBEIRO, C. *Arquitetura de computadores*. 5. ed. Rio de Janeiro: LTC, 2017.
- HENNESSY, J. L.; PATTERSON, D. *Arquitetura de computadores: uma abordagem quantitativa*. 4. ed. Rio de Janeiro: Elsevier, 2014.
- MONTEIRO, M. *Introdução à organização de computadores*. 5. ed. Rio de Janeiro: LTC, 2007.
- NÓBREGA, R. A memória. In: NÓBREGA, R. *Bem-vindo à introdução ao computador*. Paraíba: [S. n., 201-?]. Disponível em: <http://www.di.ufpb.br/raimundo/PCaFundo/memoria/memoria.htm>. Acesso em: 28 jul. 2019.
- NULL, L.; LOBUR, J. *Princípios básicos de arquitetura e organização de computadores*. Porto Alegre: Bookman, 2011.
- STALLINGS, W. *Arquitetura e organização de computadores*. 10. ed. Rio de Janeiro: Pearson, 2017.
- WEBER, R. F. *Fundamentos de arquitetura de computadores* 8. 4. ed. Porto Alegre: Bookman, 2012.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



DICA DO PROFESSOR

É uma premissa aceitável que um sistema computacional tenha uma grande quantidade de memórias com características distintas. Esse conjunto de memórias, por apresentar características diversas, entrega a um sistema uma performance equilibrada entre características de performance e custo.

Na Dica do professor, você conhecerá os tipos de memórias disponíveis em um sistema computacional e suas características.

Conteúdo interativo disponível na plataforma de ensino!



EXERCÍCIOS

- 1) **As memórias apresentam diversas características, que podem indicar diferentes parâmetros, como, por exemplo, o custo. Indique qual característica abaixo está relacionada ao tempo em que uma memória retém uma determinada informação.**
 - A) Não volátil.
 - B) Tempo de resposta.
 - C) Volátil.
 - D) Registradores.
 - E) Flash.
- 2) **A performance de uma memória é um dos parâmetros de desempenho mais importantes, uma vez que pode limitar, ou até mesmo inviabilizar, sua utilização.**

Além disso, há características que acabam por influenciar outras, em virtude de limitações ou complexidades tecnológicas. Indique a opção que representa uma característica influenciada pela performance das memórias.

A) Tempo de acesso.

B) Registradores.

C) Temporariedade.

D) Custo.

E) Capacidade.

3) A alocação das memórias é realizada constantemente em um sistema computacional, toda vez em que o sistema base ou um de seus programas são executados. Essa alocação pode ser realizada de forma dinâmica ou estática. Indique qual opção abaixo indica um código que representa uma alocação dinâmica de memória.

A) `int ValorInteiro = 133.`

B) `char ValorOpt = 'X'.`

C) `double Total = 0.`

D) `float Resultado.`

E) `vet = malloc(255).`

4) Muitas técnicas foram aplicadas às memórias em busca de melhorias nas

características. Por exemplo, ao observar a evolução dos barramentos dos módulos de memórias, pôde-se perceber que, na primeira versão da memória “DIMM”, havia apenas 30 contatos, e algum tempo depois, já havia 72. Selecione a opção que indica uma característica influenciada por essa mudança:

- A) menor custo.
 - B) maior performance.
 - C) menor complexidade tecnológica.
 - D) maior temporariedade.
 - E) menor volatilidade.
- 5) A frequência de operação dos barramentos nos quais as memórias estão conectadas está diretamente ligada à capacidade de transferência de dados que pode oferecer. Existe outra característica do barramento que também é diretamente ligada à capacidade de transmissão de dados das memórias. Selecione a opção que a indica:**
- A) O tamanho do tempo de ciclo.
 - B) Baixa capacidade de armazenamento.
 - C) Alta capacidade de armazenamento.
 - D) O número de vias do barramento.
 - E) O custo por *byte* da memória.



A forma de alocação de memória em um sistema pode afetá-lo a ponto de limitar o programa que está realizando a alocação. Atualmente, há duas formas de alocação de memórias, a dinâmica e a estática. A alocação estática é a mais utilizada em *software*; a dinâmica contém aplicações que, se faltassem, trariam muita dificuldade para o sucesso do *software*.

Veja, Na prática, como Adriana, uma desenvolvedora de *software*, decidiu por uma forma de alocação de memória para atender a alguns requisitos no desenvolvimento de um *software*.

Adriana faz parte de uma equipe que está criando um *software* de edição de arquivos texto.

A equipe de desenvolvimento está planejando alguns detalhes em relação às formas de alocação de memória, visando atender aos requisitos do *software* em questão.



Os editores de texto normalmente podem se deparar com arquivos de poucos *bytes*, mas também podem ter a necessidade de editar arquivos com muitos *bytes*.

Dependendo da forma que o aplicativo for desenvolvido, muitas situações podem ocorrer. Por exemplo, se o arquivo não for carregado em memória em sua totalidade ou no máximo possível, uma operação de pesquisa de palavras pode ser muito lenta em virtude do aplicativo ter de ler a todo instante o disco rígido, onde o arquivo pode estar armazenado.

A equipe conta com duas possibilidades de alocação de memória a serem avaliadas: alocação dinâmica e a alocação estática.

A equipe tem uma tendência a utilizar uma alocação estática, mas solicitou uma reunião com Adriana para avaliar se, frente aos requisitos, seria a melhor opção.

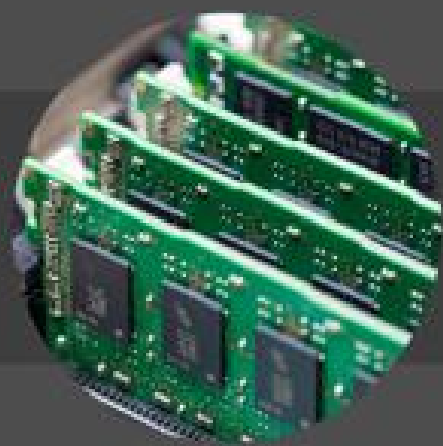
Ao analisar a situação problema, Adriana concluiu que a melhor alternativa seria a alocação dinâmica.

Essa resolução levou em conta vários fatores, sendo um deles a velocidade que o arquivo texto, uma vez carregado em memória em sua totalidade ou no máximo possível de seu conteúdo, vai proporcionar ao usuário ao realizar operações com os dados do arquivo texto em questão.



A alocação estática de memória seria uma boa opção, mas geraria desperdício em arquivos pequenos e falta de performance em grandes arquivos. Observou que teria de ser alocada estaticamente uma considerável quantia de memória, o que, em muitos casos, ainda não seria o suficiente para carregar o arquivo totalmente.

Sendo necessário, assim, utilizar um recurso de paginação onde, ao carregar e processar parte do arquivo, o restante ou um próximo fragmento seria carregado e processado assim por diante, gerando uma interação grande com o disco do sistema (que costuma ser mais lento), causando perda de performance.



A alocação dinâmica tornaria possível alocar tanta memória quanto o sistema disponibilizasse, necessitando de paginação somente como último recurso. Em arquivos menores, esta paginação não existiria, e o desperdício também não.

Após ouvirem as considerações de Adriana, foi definido que a alocação de memória utilizada no *software* de edição de texto ser desenvolvido seria a dinâmica. Após o desenvolvimento do *software* de edição de texto, foi averiguado o bom funcionamento das funcionalidades previstas.



SAIBA MAIS

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Visita à fabricação DRAM - Kingston

Neste vídeo, você poderá obter mais informações sobre a fabricação de memórias em diferentes processos existentes.

Conteúdo interativo disponível na plataforma de ensino!

O que é melhor: ter mais RAM ou ter RAM mais rápida?

Neste link, você verá uma orientação interessante sobre como analisar características de memórias RAM para escolher a melhor opção.

Conteúdo interativo disponível na plataforma de ensino!

Princípios Básicos de Arquitetura e Organização de Computadores

No capítulo 6, Memória, você poderá obter mais informações sobre os diversos tipos de memórias relacionados às suas funções em um sistema computacional.

